ATTACHMENT 1

DECLARATION UNDER 37 CFR 1.131

09/518,837

MAVENT
ASSURING REGULATORY COMPLIANCE

DECLARATION

Sir:

I, Frank D. Tuttle, do hereby declare as follows:

1. I am the sole inventor on the United States Patent Application No. 09/518,837, filed on March 3, 2000, and am familiar with its contents including its claims as amended.

2. I conceived the idea of a computer-implemented method for automated compliance auditing of loan application files in the year 1997

3. During years 1998 and 1999, I worked on the development of a prototype loan origination system that was enhanced to validate the concept of automated compliance auditing of loan application files. A major problem encountered in the development of the system was a method for (1) acquiring and maintaining the latest governmental regulatory requirements in the form of textual copies of published laws and regulations, and (2) converting the governmental requirements from a textual format into computer-implemented rules that could be automatically executed by a computer to determine if a loan file was in compliance with the regulatory requirements. A workable solution to this latter problem (2) was implemented in April, 1999 and subsequently tested and demonstrated. A copy of the software documentation for the rule builder software is included in Exhibit A of this Attachment 1. This documentation is a part of the

3 Park Plaza, Suite 700
Irvine, CA 92614
ph > 949.474.4700
fax > 949.474.4701
www.mavent.com

documentation for the software versions 2.32a through 2.40a that was last updated on September 30, 1999. Although some sections were incomplete, salient sections of this software documentation in attached to this declaration as Exhibit A. Exhibit A includes a Table of Contents and descriptions in SQL syntax for Installation, Rule Builder and Rule Library's, License Audit, Section 32 (high cost mortgage), Table Field Descriptions, and Software Version Changes. The Rule Builder describes the implementation of the rule processing data via relations data model. The Rule Library's provides reference the "State Fee rules". The License Audit describes the state audit evaluation rules. Section 32 describes processing for high cost mortgages. The Software Version Changes describes the software version changes from version 2.32a to version 2.33e of September 30, 1999.

4. On December 8, 1999, I sent a loan application compliance auditing system disclosure document to the intellectual property law firm of Taylor Russell & Russell, PC and requested that a patent search be conducted. A copy of this disclosure document is attached to this declaration as Exhibit B.

5. On December 16, 1999, I received a copy of a patent search results document from Taylor Russell & Russell, PC that was based on the disclosure document that I sent them. A copy of this patent search result document is attached to this declaration as Exhibit C.

6. On December 22, 1999, I received an intellectual property representation letter from Taylor Russell & Russell. PC, which I signed and returned with a deposit fee and instructions to proceed with the preparation and filing of a utility patent application for my automated compliance auditing of loan system. On December 28, 1999, I emailed the invention disclosure document shown in Exhibit D to Taylor Russell & Russell, PC. On February 14,

2000, I provided an updates version of the invention disclosure document to Taylor Russell & Russell, PC.

7. On February 17, 2000, I received a first draft of a patent application for my invention from Taylor Russell & Russell, PC. I reviewed the draft and returned it with my comments on February 28, 2000. Taylor Russell & Russell, PC subsequently filed a utility patent application for my invention as U.S. Patent Application No. 09/518,837, entitled LOAN COMPLIANCE AUDITING SYSTEM AND METHOD on March 3, 2000. The declaration filed with the patent application included the statement that I had reviewed and understand the contents of the identified specification, including the claims.

8. That between December 8, 1999 and the filing date of March 3, 2000 for U.S. Patent Application No. 09/518,837, I was diligent in perfecting a constructive reduction to practice of my invention.

9. That all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements are made herein with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Title 18, § 1001 in the United States Code, and that any such willful false statements may jeopardize the validity of the application or patent issuing therefrom.

Date: April 6th, 2007

Signed: Frank D Tuttle

Frank D. Tuttle

EXHIBIT A

Table of Contents

1.0 Installation

Foresight Mortgage Systems, LLC.

last update: 9/309/99 (v2.40a)

This information is organized to provide details on:

1. The setup of the system
   An explanation for the User Interface (UI) screens
2. Trouble Shooting

Additional information is provided that brings together the required system design relationships and the effect upon the data.

3.0 Setup

3.1 Rules Building

Rule Builder

last updated: 4/1/99

Components:
The rule building functionality is built around the data table:
tbl_su_rule_library_master

lib_id	Long	Unique record ID
lib_name	varchar	Description for the record
lib_ret_tbl	varchar	Name of table that rule id will be returned to from the library.
lib_ret_id	varchar	Name of first key field that may need to be updated if rule builder adds a new record in the lib_name table.
lib_ret_varfld1	varchar	Second key field.
lib_ret_varfld2	varchar	Third key field.
lib_ret_varfld3	varchar	Fourth key field.
lib_ret_varfld4	varchar	Fifth key field.
lib_ret_tblid	varchar	Field to return the rule to.
lib_ruleDB_id	varchar	Unique ID field in the master data table of the library where the rules are stored.
lib_ruleDB_name	varchar	Master rule table name.
lib_ruleDB_desc	varchar	Master rule description field to use for each rule.
lib_ruleDB_fldsql	varchar	Master rule field containing the SQL string representing the rule.

lib_build_qry	logical	Name of the table or query that is used to build the rules.
lib_sqlterm_chr	logical	Boolean indicating whether the rule builder will insert a termination character after each sub rule. This is required for parsing non SQL where clause rules.
lib_single	logical	True if this rule instruction set can only return one rule to the requesting form as opposed to many. Some rule relationships support many rules (state limits) while others support only one rule per element (documents).
lib_edit	logical	True if the passing of a rule back to the requesting form table requires a recordset .Edit to update otherwise an .AddNew update will be performed.
lib_tblqry	logical	False if the print logic is built from the framework of a query (RowSourceType property in the list box is set to "Field List"). True if the rule is built from the contents of a table (List box RowSourceType is set to table/query).
lib_sqlbldr_msg	varchar	Rule building message.
timestamp	datetime	

Forms Used:

frm_rule_library: This is the library display and test form. This form interprets the request when it is opened, gets all of the required instructions from the master table and prepares the interface for use. This form will also send any required instructions on to the SQL builder form if the user requests to create a new rule.

frm_rule_sql_builder This is the SQL or interpreted rule building form. This form will provide the interface to the user to build rules. The list box on the left of the form will display all applicable data elements that can be evaluated for this rule library. Certain predefined data elements have lookup values that this form interprets and displays to the user. Some libraries allow the math operators "IN" and "NI" which are IN and Not In respectively.

IN:
state in ("CA","ID","NV","CT","NH")

provides an abreviated rule versus the more traditional

state = "CA" or state = "ID" or state = "NV" or state = "CT" or state = "NH".

NI works just the opposite in terms of the logic but exactly the same way in terms of implementation.

As soon as one logical expression is transferred down to the "SQL Logic Text" box the user can start defining additional logical items and increase the complexity of the expression evaluation. The down arrow appends the users selection from the Data, Eval and Value selections down to the SQL Logic Text. An "AND" joiner is included by default unless the user selects and "OR" joiner. Parenthese can be used and should be used when "AND" and "OR" joins are combined in

complex logical expressions. Once completed the expression can be transferred back to the library form and tested there.
The SQL Logic Text box allows editing.

Rule Library's

Last Update: 5/17/99

There are several rule libraries maintained in Foresight:

- Print rules (interpreted)
  State Fee rules (SQL syntax)
- Warehouse line rules (SQL syntax)
- Price Adjustment rules (SQL syntax)

The master table holding the rule instructions is: tbl_su_rule_library_master.

lib_id	Int	Rule Table Primary key field
lib_name	varchar * 25	library name to operate on
lib_ret_tbl	varchar * 25	name of table to write to
lib_ret_id	varchar * 35	name of the field to use for the primary key ID when processing user selections
lib_ret_varfld1	varchar * 25	optional additional data primary key 1 to write to table along with rule id (fld name)
lib_ret_varfld2	varchar * 25	optional additional data primary key 2 to write to table along with rule id (fld name)
lib_ret_varfld3	varchar * 25	optional additional data primary key 3 to write to table along with rule id (fld name)
lib_ret_varfld4	varchar * 25	optional additional data primary key 4 to write to table along with rule id (fld name)
lib_ret_tblid	varchar * 25	Replacement field for the autonumber (autonumber - primary key)
lib_ruleDB_id	varchar * 25	Name of the rule id field
lib_ruleDB_name	varchar * 50	Name of the rule DB table
lib_ruleDB_desc	varchar * 75	name of field to write Description to
lib_ruleDB_fldsql	varchar * 150	name of field sql to write sql to
lib_build_qry	varchar * 25	name of the query to build logic from
lib_sqlterm_chr	bit	Termination character at end of sql string (used in print logic as example). Include a termination character - used only for interpreted rules that require a function to parse out the logical elements.
lib_single	bit	Limit user selections to one item only
lib_edit	bit	Yes if this is to edit an existing record
lib_tblqry	bit	True if the query definition will be used to build the rules, false if the data returned by a query will be the source of the rules building.

lib_sqlbldr_msg varchar * 250 message to the user on the SQL bulder form

3.5 Audits

3.5.1 License Audit

Function description:

```
Function gTestLicenses(SrcID As Long, lnSet As tLoanSet, retMsg As String, pLicID As Long) As Boolean
'-----------------------------------------------------------------------------------
' Tests the Originators licensing for the LnSet loan file.
'
' Parameter Calls:
' Parameter options
' SrcID Source_id for license to audit
' lnSet specific loan file data
' retMsg Return message
' Dev Name: Date: Comments:
' Frank Tuttle 3/4/99 10:29:21 AM Frank Tuttle
'-----------------------------------------------------------------------------------'
'qry_statelicense_orig --- filter by state_ini and source_id
'qry_statelicense_lender
```

Get the state licensing data for the state and source company
If not data then
Add Message "*** Check the state license for this Source since no records were returned to test due to no data having been setup in the state license tables in Source Maintenance. ***"
gTestLicenses = True
Exit this Audit
If there is data then
Test: If the license is not perpetual and the review date is less than todays date then need to review.
Add Message = "Ths State license is perpetual and needs to be reviewed."
Test: If the state requires a license and there is not a perpetual license and there is no exemption then
Test: If License Expiration is less than todays date then Add Message "Check the state license expiration date and setup data."
Test: If the state requires a license and there is not a perpetual license and there is an exemption then
Test: Check for the appropriate exemption type
If there is an exemption then Add Message: "A state license is required except this source has been tagged to have a " & ptstr & " exemptiion."
Test: If no license is required then Add Message: "No state licensing is required."
Test: If license in state is active. If not Add Message: "Data indicates that this Source is not currently active to originate loans in '" & lnSet.PropState & "'."
Test: Get all licenses for this state and see if one works; If not then Add Message: "The State License Rule {" & rst!pl_desc & "} for the license " & rst!stlic_desc & " Failed."
Return the license ID that works.

3.5.2 Lock Audit

Function description:

```
Function gAuditLock(LN As String, pfileCorrect As Boolean, rtnMsg As String) As Boolean
'-----------------------------------------------------------------------------------
' Audits the loan lock
'
' Parameter Calls:
' Parameter options
' LN Auto Correct data when possible if True
' pfileCorrect If True, the loan file will be corrected to match the lock
' Rate, Loan Amount, Program Code, Lien, Fees = 801, 802, SRP and Rebate
' rtnMsg The return message of findings, this message will be logged if the pfileCorrect is set to true.
'
' Dev Name: Date: Comments:
' Frank Tuttle 1/12/99 11:21:54 PM Frank Tuttle
'-----------------------------------------------------------------------------------'
```

Test: Loan File Loan Amount against Locked Loan Amount
Test: Loan File Loan Interest Rate against Locked Loan Interest Rate
Test: Loan File Loan Program Code against Locked Loan Program Code
Test: Loan File Loan File Fees Discount, YSP, SRP and Rebate against Locked Loan Pricing Terms
Test: Loan File Loan Margin against Locked Loan Margin

3.5.3 Program Parameters

Function description:
Function gAuditProgParams(LN As String, rtnMsg As String) As Boolean
'--
' Audits the loan Program code to validate that the loan application meets the defined parameters for the program code. Function will First evaluate if there is any data to
' test before rendering an opinion.
' Parameter Calls:
' Parameter options
' LN Auto Correct data when possible if True
' rtnMsg The return message of findings, this message will be logged if the pfileCorrect is set to true.
'
' Dev Name: Date: Comments:
' Frank Tuttle 1/13/99 11:21:54 PM Frank Tuttle
'--'
Checks Loan:
LTV
Program Code
CLTV
Gross Loan Amount
Lien
Occupancy
Property Type
Purpose
Doc Level
Mortgae Applied For

3.5.4 Source Audit

Function description:
gAuditSource(SourceID As Long, LN As String, stateini As String, lntype As Integer, lngApprovalID As Long) As Boolean
Audits the Originator of the loan, Uses qry_source_audit as data source for the audit
writes results to a conversation logging routine.
' Parameter Calls:
' Parameter options
' sourceid Source to audit
' LN Loan number - to write the conversation log results to
' state_ini state loan resides in for audit
' lnType FHA need HUD numbers (1;VA;2;FHA;3;Conventional;4;FmHA;5;Other)
Audit Checks:

If No Source Data then sends a message to the user if there is not any data and then Exit Function.
If Data for Source Exists then
Get all data for that state.
If No State Data
then send message "The source maintenance data for '" & stateini & "' is incomplete and needs to be setup."
If State Data then
Check Licenses: gTestLicenses gLoanFile.SOURCE, gLoanFile, strMsg, pLicID
'check the source company data which is independent of state
Check Financials Date
If the loan is FHA Check to see if there is a HUD Number.
Check for a tax ID.
Check for conv QC.
Check for Status (if not approvaed then log the status).
Check the approval date.
Check the inactive date.
Check the active check box.

If there is a failure then send the failure message to the conversation log.

3.5.5 State Compliance Limits

Function description:
Function gAuditStateLimits(LN As String) As Boolean

```
'------------------------------------------------------------------------
' Audits the loan to the state specific limits. Use qry_statelimits
'
' Parameter Calls:
' Parameter options
' LN Pass in the loan number
'
' Dev Name: Date: Comments:
' Frank Tuttle 9/19/97 3:21:54 PM Frank Tuttle
'------------------------------------------------------------------------'
```

Check if loan is open end or Closed End
Check if Lender is the Originator
Test State Licenses for
Lender or Lender & Originator

Terminology"
'stRuleApply the rule on the state limit - whether or not this state limit should be applied to the given loan.
'stLicenseApply the rule on the license for either orig or lender (this test may have already been run for the stlic_id
'strule_sql this is the rule or limit representing the state restriction
'------------------------------------
'== Test the individual rules here == using qry_state_sql_limits_audit
'------------------------------------
'strule_sql is the field that the filter will use
'strule_desc is what needs to be appended to the message string on fail
'All rules are written in a way that a failure occurs if the filter returns a record. (that rule fails)
Check all rules in a loop here, add failure message if rule fails.

Section 32

last update: 9/16/99
The interpretation of the Section 32 High Cost Mortgage requirement is crucial to acurate limit calculations. The following notes should help document how Foresight processes the data to support the calculations.
Critical Issues:

- Selecting the proper index to use for the APR test: The index to use is the closest Treasury Instrument in maturity to the maturity of the loan. So a 144 month loan will use the 10 Year Note since its maturity is closer to 120 months thain it is to the 20 Year Bond of 240 months. When the term of the loan is exactly between the terms of the Treasury Instruments then the requirement is the use the lower of the two index values. So a 180 month loan will evaluate the index values for the 10 and 20 year instruments to get the lower value to use in the 10 point test. Also the term of the loan is the maturity term if the loan is a balloon loan.
- Date of the index value to use: Foresight will use the index in effect of the week that the 15th of the month is in. So Foresight looks for a "Section 32 Date" in the system between the 15th of the month and the 22nd of the month. The "Section 32 Date" auto calculates the date that he index should be available in publication whenever a date is entered in that field for a given month. So if a user enters 8/1/99 for instance the system will automatically change the date to 8/16/99 which is the date the the index for the week that the 15th is in will be available.

D. Table Field Definition

Table: Tbl_Idf_loaninit Last Update: 8/27/99

This is the master table for all loans in Foresight. Deleting a record from this table will cascade through all loan tables and delete all loan data. Renaming this loannum in this table will also cascade the update through the entire loan record.

Field Name	Data Type	Where Used	Description
loannum	varchar (15)		Unique Loan Number provided by Lender
appdate	datetime		1003 Application date
submitdate	datetime		Date file received by Lender
entrydate	datetime		Date File entered into computer
statuscode	integer		Current status code
statusdate	datetime		Date of current status change
statusby	integer		Employee id of person who updated the status
statusactive	varchar(1)		tbl_su_status_activity
reqfunddate	datetime		Used by processing as a target date for funding
wkschid	integer		ID for the work schedule assigned to this loan
init_doc_prntdt	datetime		date docs printed
init_s32_noticedt	datetime		date section 32 notice signed
init_doc_signdt	datetime		date docs are signed
init_fund_dt	datetime		date loan funded
init_actioncode	tinyint/byte		HMDA action code
init_actiondate	datetime		HMDA action date
init_purpose	tinyint/byte		HMDA purpose
init_bldr_constr	boolean/bit		If this is a builder sales or constr loan then do not want to include the actual date events in the calcs nor delete the loan due to inactivity.
init_est_close	datetime		Estimated close date for the loan. Very important for builder loans.
init_record_dt	datetime		Recording date for loan
tsch_id	integer		Unique identifier of the team assigned to the loan.
init_temp_loan	boolean/bit		Is this a test/temp loan
init_temp_desc	varchar(35)		the description for the temp/test loan
init_early_discl_date	datetime		Early disclosure date
init_doc_exp_date	datetime		Doc Exp Date - Use this field to specifically identify when a doc set will expire - can use to notify the signer.
init_last_sign_dt	datetime		Last Sign Date - use to notify the signer when the last date the docs can be signed is.

init_earliest_fund_dt	datetime	Earliest Fund Date - use to notify the outside closing agent as to when the earliest funding date is.
init_last_fund_dt	datetime	Last Fund Date - use to notify the closing agent as to when the last funding date is.
init_loan_disb_dt	datetime	loan Disbursement Date - this is the date when funds are disbursed. For VA loans this will become the interest from date for prepaid interest.
tsch_contact_id	integer	The employee id of the team contact, usually the assistant underwriter.
init_return_hrs	tinyint/byte	Hours Escrow needs to return docs prior to funding
init_return_cutoff	varchar(8)	time of day Escrow needs to return docs prior to funding
init_earliest_sign_dt	datetime	Earliest Signing Date
init_ship_date	datetime	date loan shipped
init_corp_sent_to_rcvd_date	datetime	date the corporate assignment sent
init_file_name	varchar(25)	a name that represents the file for reporting (if there are many borrowers - this could be the identifying borrower that is used for reporting purposes)
init_status_comment	varchar(40)	A shipping comment field
init_sold_date	datetime	date loan was sold
init_corp_sent_to_rcrd_date	datetime	
init_file_name	varchar(25)	System auto-inserts a borrower name as a reference for the file here for ease of reporting.
init_trans_proc_req	varchar(1)	
init_export_ver_ind	varchar(1)	
timestamp		timestamp field
init_mers_min	varchar(18)	Mers MIN number
init_mers_status	varchar(3)	Mers File status
init_mers_consol_min_no	varchar(18)	MERS consolidated MIN no
init_mers_rae_min_no	varchar(18)	MERS RAE number
init_mers_newserv_org_id	varchar(7)	MERS New Servicer Org ID
init_mers_newsubserv_org_id	varchar(7)	MERS new subservicer Org ID
init_mers_newdoc_cust_org_id	varchar(7)	MERS new Doc Custodian Org ID
init_last_cancel_dt	datetime	Last day borrowers can cancel loan docs for ROR.
init_prequal	bit	True if loan is a pre-qual only (no HMDA reporting and exclude from some reports)

init_package_sent_date	datetime	Date 1003 application was mailed to borrower.
init_package_rcvd_date	datetime	Date 1003 application was received from borrower
init_mornet_casefile	varchar(30)	DU case file number
init_mornet_credit_ref	varchar(15)	Credit Supplier credit report ref number
init_mornet_agency_name	varchar(35)	Credit report agency name
init_mornet_agency_id	varchar(3)	Credit report agency name
init_report_type	varchar(2)	DU Credit type code

9.0 Software Version Changes

2.32a

The major change in version 2.32a was the splitting of MtgUser into two distinct applications (MtgUser and MtgRpts). MtgUser is partially securitized and mtgRpts is not securitized. Foresight licensees will unlimited access to this application to change, edit, add, delete reports to support their own reporting needs. With regards to the code modules:
New code and modules may be added but **don't make any changes to existing code** as this code is also used by MtgUser as a library.

2.32b

General

- If you enter a zip code that has more than one city attached to it, it will now pop-up a box that will allow you to select the correct city.
- The loan renumber process is now working.
- Underwriting conditions will longer re-sort until you exit the underwriting screen.
- If the loan is a pre-qual, there is a checkbox in the status screen. When the loan changes from a pre-qual to a "real" loan, uncheck the pre-qual box. (This box can also be checked when the loan is initially entered in the Loan Registration screen).
- New fields have been added to the Loan Terms screen for construction loans.
- The "Postal City" and "City and Area" fields have been added to the Setup Details screen in the Processing Mode. This is so if you are performing HMDA on the processing side and you have problems with msa/county code, you can verify the address without going to the Doc Mode. The way these fields must be used is (1) if there is no area do not put anything in the "City and Area" field or (2) if there is an area type the city and area as you want it to appear on the documents in the "City and Area" field. For example type Los Angeles (Whittier Area) or Rancho Santa Margarita

Area and that is the text that will appear in the city field on the documents. You must also put the postal city in the "Postal City" field. So, if Whittier is the where the post office would send mail, put in Whittier; if Rancho Santa Margarita is where the post office would send the mail, Rancho Santa Margarita is what you need to put in the "Postal City" field. The reason this is so important is because HMDA is being based on the "Postal City" field.

Reports

- There is a new print menu that appears when you preview reports. When you click on the Print button it drops down a menu list you can select print, page setup or close. Export will eventually be an option available through this menu.
- The QC Monthly Funded Loan Detail by Branch report now includes the submit date.

Docs

- On government loans the dollar portion of the MI is now defaulting into the Financed MI field and the cents remain in the Cash MI field.
- Site condos will now check the condo box on the deed of trust.
- The new PMI disclosure is now included in the doc sets.
- Data Entry/Processing
- When entering an address you will not longer get an error if entering a street name with an apostrophe in it.
- There is now a check box for if the loans have impounds and a check box if the loan is a pre-qual. Simply check the appropriate box if it applies.

Underwriting

- Cash deposits from all borrowers will now appear as the down payment in the Loan Terms screen.

Compliance

- When fee names are changed in the HUD fee preset, the new fee name will now be reflected in the loan file when the preset is selected.
- The State Licenses and Rules report is now available.

Shipping/Accounting

- There is now a "Misc Fee" in the Purchase Advice screen.
- You can now add a fee comment longer than 30 characters.

2.32c

Wholesale/Broker Administration

- Foresight now has the ability to change the account executives on an individual loan basis. This is so that if an account executive is changed in the master setup, the account executive assigned on a loan level basis will not change. This is because if an account executive changes or leaves, he/she still need to be compensated on loans currently in the pipeline. However, it is important to realize if all loans in the pipeline are being changed to the new account executive, you will have to go into each loan to make the change.

General

- The Broker/Seller screen has been fixed.

Docs

- The payment amount for balloon loans has been corrected.

Reports

- The Lock Report - Admin has been fixed corrected.
- The Cancelled Loan by Loan Officer has been corrected.
- The Broker Report has been corrected.
- The Application Detail by Acct Exec is currently not working (it will not return all the data).

2.32d

Secondary Lock Roll calculation - Corrected a bug when a holiday falls on a Monday and the lock roll is forward.

BuyDown auto-calc of start rate and interest cost on fees screen - Fixed a bug that was introduced in 2.30.

2.33a Release 8/26/99

SQL Database

Fields:

- **Add** tbl_ldf_loaninit.init_package_sent_date = datetime; *(the date a loan application package as sent to a prospective borrower)*
  Add tbl_ldf_loaninit.init_package_revd_date = datetime; *(the date a loan application package as received from a prospective borrower)*
- **Add** tbl_gprogramdefs.def_min_term smallint def = 0; *(for the program note setup, this is the minimum allowed term in months for the note to be used)*
- **Add** tbl_gprogramdefs.def_max_term smallint def = 0; *(for the program note setup, this is the maximum allowed term in months for the note to be used)*

The tbl_ldf_trw110 tables are the header data for credit pull history. Foresight can store an unlimited number of credit reports that were imported and then the user can decide which report is loaded into the loan application using the Credit Manager.

- **Add** tbl_ldf_trw110.trw100_mornet_casefile varchar(30); *(DU return file casenumber to be used for resumissions (for future use))*
  Add tbl_ldf_trw110.trw100_mornet_credit_ref varchar(15); *(Credit ref supplied by DU approved credit suppliers in lieu of DU ordering report)*
- **Add** tbl_ldf_trw110.trw100_mornet_agency_name varchar(35); *(Credit report data supplier name)*
- **Add** tbl_ldf_trw110.trw100_mornet_agency_id varchar(3); *(Credit Supplier ID (DU))*
- **Add** tbl_ldf_trw110.trw100_report_type varchar(2); *(DU code for the combination of repositories used)*
- **Add** tbl_ldf_loaninit.init_mornet_casefile varchar(30); *(DU return file casenumber to be used for resumissions)*
- **Add** tbl_ldf_loaninit.init_mornet_credit_ref varchar(15); *(Credit ref supplied by DU approved credit suppliers in lieu of DU ordering report)*
- **Add** tbl_ldf_loaninit.init_mornet_agency_name varchar(35); *(Credit report data supplier name)*
- **Add** tbl_ldf_loaninit.init_mornet_agency_id varchar(3); *(Credit Supplier ID (DU))*
- **Add** tbl_ldf_loaninit.init_report_type varchar(2); *(DU code for the combination of repositories used)*
- **Add** tbl_su_svcr_details.svcr_street2 varchar(35); *(added a second line for Investor Servicer's)*
- **Add** tbl_su_investors.Inv_Address2 varchar(50); *(Added a second address line for Investors)*
- **Edit** tbl_su_investors.Inv_Phone varchar(15); was 14; (Added an extra character for the phone extension)

Edit tbl_su_investors.Servicing_Phone varchar(15); was 14; (Added an extra character for the phone extension
- **Edit** tbl_su_investors.Ins_Phone varchar(15); was 14; (Added an extra character for the phone extension)
- **Edit** tbl_su_sver_details.sver_ph varchar(15); was 12; (Added 3 extra characters for the phone extension)
- **Edit** tbl_su_sver_details.sver_toll varchar(15); was 12; (Added 3 extra characters for the phone extension)
- **Edit** tbl_co_providerdatabase.providerinternetaddress varchar(50); was shorter; *(Lengthened the loan provider email address)*
- **Edit** tbl_su_rules_fld_queries.fld_query varchar(255); was 35 in length; *(lengthened this field which is used to store the name of the lookup table/query to be used to present the options to select in the rules builder utility. Now Foresight will allow a field list string to be used in addition to a table or query. A field list string is the value list of choices such as: @1;Conforming;2;FHA;3;VA;4;Other. A field list needs to be preceeded by the "@" character so Foresight knows to present the data as a list as opposed to trying to find a table or query.)*
- **Edit** tbl_co_sources_approvals.tpo bit def = 0; *(ensure that there is a default value for this bit field)*

Stored Procedures:

sp_qry_rpt_soldsett_233a *(for new report rpt_sold_settled_by_date)*
sp_qry_rpt_rcvd_noisett_233a *(for new report rpt_sold_rcvd_notsent)*
sp_qry_rpt_miss_docs_233a *(for new report rpt_sold_miss_trail_docs)*
sp_qry_rpt_sett_by_inv_233a *(for new report rpt_sold_sett_by_inv)*
sp_qry_rpt_loans_shpd_233a *(for new report rpt_loans_shipped)*
sp_qry_rpt_wrhs_info_233a *(for new report rpt_wrhsed_loans_not_soldsett and rpt_shipped_loans_not_soldsett)*
sp_NewLoanInitMT_233a *(added two new fields that the new_loan_init process performs)*
sp_global_structure_233a *(added program init amortization to the global structure to support interest only amortization during an initial ARM period)*

Applications

MtgUser.MDB and PreFrms.MDB

Overview

Bugs

- **Issue**: Buydown recalc. A recent release affected the way the buydown is updated between the terms and fees screen so that there were some scenarios when the buydown was not getting updated properly.
  Response: Fixed
- **Issue**: Holiday lock roll falling on a Monday which is also a holiday not rolling correctly.
  Response: Fixed.
- **Issue**: AGI income transfer generating an error.
  Response: Corrected the spelling in code of the name of the object that hte loan income will be sent to.
- **Issue**: Marketing Plans Add (Secondary). Adding a new marketing program through the form called from the trade maintenance screen generates an error.
  Response: Found and fixed.
- **Issue**: Note Plans Add/Edit (Secondary). Making changes to Note Plans would sometimes generate a Write Conflict error message.
  Response: Rewrote the save processes and Write Conflict message has not appeared.
- **Issue**: In House Program Maintenance (Secondary). Double clicking on the right list box when there nothing was in the list produced an invalid use of null error message.
  Response: Added code to ignore user requests when there was nothing selected to transfer.
- **Issue**: Payment Stream. This was fixed in an earlier 2.32 version but will be

documented here. The last payment was a little higher than expected. Because of rounding odd cents, the last payment should be within one cent times the number of months in the entire payment stream (plus or minus). So if there are 360 months, the last payment should be within $3.60 of the prior payment.
Response: It was found that the last payment was calculating in error and was adding interest on the interest of the last payment.

- **Issue**: Note Maintenance GPM fields allowing user input for non GPM loans.
  Response: Fixed.
- **Issue**: MI plans transfering data to the loan file on the Fees screen.
  Response: Working. Added code to transfer MI pricing.
- **Issue**: Note Plan form; the min life rate rounds to a whole number.
  Response: Fixed.
- **Issue**: Note Plan; qualifying info not working.
  Response: Foresight is currently not using the qualifying information, it may be used in a future release. It was used for a qualifying program that is no longer used.

Enhancements

- **Issue**: From Loan Finder could not search for "Agency Case Number".
  Response: Added a check box next to the Loan Number text box on the top of the form that when checked will go to the loan with the "Agency Case Number" that is entered. The list box will not display "Agency Case Numbers" so the case number must be manually entered.
- **Issue**: Combo boxes: Mtg Type, Doc, Occupancy, Property, Purpose need to be ordered alphabetically (on New Loan Init and Terms screen).
- **Response**: List values now appear ordered alphabetically.
- **Issue**: Provider email address is too short.
  Response: Length has been lengthened to 50 characters.
- **Issue**: Want to be able to filter fees by Active and Inactive fees.
  Response: Added check box's for displaying fees by Active, Inactive and Both.
- **Issue**: Provide easy access for users to get to a calculator.
  Response: Added the underwriting calculator and the standard windows simple calculator to the menu that is open when a loan file is open.
- **Issue**: The underwriting calculator was not documented and a little difficult to understand.
  Response: Added a second tab to the form and included a help dexscription of how to use it and enhanced the use so that when a scenario is displayed it can easily be modified so that the user can only make one change to do what-if changes and re-calc the results.
- **Issue**: Because the term of the loan was set in the note maintenance, it required many notes just to support a different term.
  Response: Added a min term and a max term to the note plan and now use the original term as the default value. The loan now has two fields being the amortization term and the loan term (if a balloon) that have been put on the Terms screen. The system will check and verify that the user is entering an acceptable term when either the loan program is changed or the term is changed. If the user tries to set the loan term to less than the minimum allowed by the note plan the term will be set to the

minimum value allowed (same for max value). The loan term will allways be the note repayment due term for balloon loans and it will be set to the amortization term for all other loan types.

- **Issue**: There are some new ARM loans that have an interest only payment during an initial period.
  Response: Note plan maintenance: Added a user selectable field for ARM notes that have initial periods other than the standard changes (such as 3/1, 5/1, 7/1 and 10/1 ARM's) that accomodate the initial period being interest only for the payment calculations. The APR and payment stream calculations were modified to accomodate a loan set up this way. Also added the initial amortization to the print rules to accomodate special notes and riders for these types of loan programs.
- **Issue**:
  Response:
- **Issue**:
  Response:
- **Issue**:
  Response:

MtgRpts.MDB
Overview

A number of new reports were added to the base code. Some of these reports have criteria set up for them which means that some of the report criteria tables need to be updated to fully use these reports.

Startup
Overview
Two new fields were added to startup.
tbl_application_file.file_registerdll, boolean value; (used to instruct the startup code to automatically register the component if this boolean value is true. Startup can deploy and register activeX components using this feature without a dedicated setup.exe.)
tbl_application_file.file_operation_code, byte value; (for future use).

2.33b 8/29/99

SQL Database
Fields:

- **Edit** tbl_co_providerdatabase.contactsemailaddress varchar(50); was 25; *(Lengthened the loan provider contact email address)*

Applications
MtgUser.MDB

Bugs

- **Issue**:Creating a new loan caused an error.
  Response: Corrected a misspelling in the call to a stored procedure to save loan data.

Enhancements

- **Issue**: Request to add tag support to the loan instructions similar to the way UW conditions work.
  Response: Added tag replacement support to the instructions with the limitation that only one tag will be processed per instruction. Also added many new tag options. This release allows many tags per Instruction or UW Condition unlike the replaced functionality that only allowed one tag per UW Condition. Added a popup message displaying all of the tags with an explanation for each. Now code will convert most tag values from a code to a text string. Replacements tag values will be formatted in most cases for insertion into the string for readability.
- **Issue**: Email address in Providers screen is too short (25).
  Response: Modified Contact and Provider email address to be 50 characters and widened the data entry area.
- **Issue**: Need an additional address line in Investor data screen (PrcFrms) and allow 5 characters for the phone extension.
  Response: Completed. Also extended the extentions for the servicing providers and added an additional address line.

2.33c 9/7/99

SQL Database

Fields:

- **Edit** tbl_ldf_borrower Changed the default values of bcitizen and bresidentalien from 0 to 1

Applications

MtgUser.MDB

The following queries need to be run to update the system

qry_update_newloandates_233c : Updates the date/time stamps for fields submitdate and appdate in tbl_ldf_loaninit to contain only dates with no times.

Stored Procedures

sp_qry_rpt_app_detail_by_ae : Added to the where clause statlvl <> 23 and aliased submitdate as submit_date

sp_qry_rpt_app_detail_by_lo : Added to the where clause statlvl <> 23 and aliased submitdate as submit_date

sp_qry_rpt_sett_by_inv_233a :Added the field trade_partyname so the criteria would work correctly

sp_qry_rpt_miss_docs_233a :Added the field trade_partyname so the criteria would work correctly

Triggers

tbl_ldf_borrower_Dtrig: Created a new trigger for when a borrower or coborrower is deleted

Bugs

- **Issue**:1008 not calculating the correct BE ratio for Investment property transactions.
  Response: Added two new queries and remapped Doc Mrgflds field ID 1468 to the new query and field qry_DocMap_1008OtherMP.aomp.
- **Issue**:Section 32 periodically incorrectly calculates the dates in Doc Dates.
  Response: Identified and corrected the routine that moves to the next available date.
- **Issue**: If a user modifies a loan balance or payment amount in the liability screen for a RE item that has been attached to a Investment piece of Real Estate, the system will not automatically recalculate all of the income numbers.
  Response: Any modification to a Mortgage/Lien that has been assigned to an Investment property must be modified in the Schedule of Real Estate screen. The system will no longer allow a mod to the balance or payment in the Liability screen for this type of item.
- **Issue**: Subordinate financing would peridically incorrectly calculate when there was a third position loan.
  Response: Found the offending code and corrected it.
- **Issue**: HUD 901 (prepaid interest) would calculate off the fund date if there was a fund date (for TIL).
  Response: Changed to use the dates on the Fees screen (from date and to date) for the interest calculation.
- **Issue**:Past Employment income showing up on the wrong line.
  Response: Modified the array numbering on the CDF file for the 7300 form. There is no code change for this one, just use the new CDF file.
- **Issue**: Lock Report Admin charts were not showing up.
  Response: Changed the queries being referenced in the charts to the query being used with the report.
- **Issue**: Interest was not being calculated correctly when program was changed.
  Response: init_fund_dt is no longer being set to from date.
- **Issue**: In the loan finder the purpose search field did not return correct results.
  Response: Changed the id order of the purpose
- **Issue**: When editing a fee in the hud fee presets form the gl account id field contained invalid data.
  Response: This data is not necessary for hud fee presets so that field is no longer visible.
- **Issue:** On the GFE screen when changing the Cash MI to Financed MI the fees based on a percentage of the loan were not changing to be based upon the new gross loan amount.
  Response: Fixed

Enhancements

General

- Funding / Warehouse Wire form defaults as wire option
  Double clicking in the application date and submission date fields of the new loan init form gives only the date and no time stamp.

2.33d 9/7/99

SQL Database

Fields:

- **Edit** tbl_ldf_tilgfe.motaxespd1 varchar(8); was 3; *(Lengthened the accomodate the year as well as the month)*
  Edit tbl_ldf_tilgfe.motaxespd2 varchar(8); was 3; *(Lengthened the accomodate the year as well as the month)*
- **Edit** tbl_ldf_tilgfe.motaxespd3 varchar(8); was 3; *(Lengthened the accomodate the year as well as the month)*
- **Edit** tbl_ldf_tilgfe.motaxespd4 varchar(8); was 3; *(Lengthened the accomodate the year as well as the month)*
- **Add** tbl_ldf_MtgType.mtg_mailing_address1 varchar(50); *(Added this field for use as the Primary Mail Address for Correspondence to the borrower.)*
- **Add** tbl_ldf_MtgType.mtg_mailing_address2 varchar(50); Ibid
- **Add** tbl_ldf_MtgType.mtg_mailing_city varchar(30); Ibid
- **Add** tbl_ldf_MtgType.mtg_mailing_state varchar(2); Ibid
- **Add** tbl_ldf_MtgType.mtg_mailing_county varchar(40); Ibid
- **Add** tbl_ldf_MtgType.mtg_mailing_zip varchar(10); Ibid
- **Added** timestamp timestamp to all tables that it did not exist in. See related ODBC Driver Note 1.06 SQL Driver and Write Conflict Error .

Applications

MtgUser.MDB

Update Queries to Run:

- update_233d_dueprior; *Resets this field from its prior usage to false and will use this field to indicate if an impound item has been paid prior to close.*
  update_233d_feepcntof; *Updates this field from a null to a 0.*
- update_233d_mostaxespd1; *Updates this field from a date format of "Nov" to "Nov/1999" to include the appropriate year.*
- update_233d_mostaxespd2; *Ibid*
- update_233d_mostaxespd3; *Ibid*
- update_233d_mostaxespd4; *Ibid*
- update_233d_mailaddress; *Updates the mail address to the correct borrower address.*
- update_233d_10Yindex_uses32; *Insurance that the 10 year index is considered with section 32 calcs.* Section 32
- update_233d_appr_not_acc_den; *Allows the Approved Not Accepted status to drive the Adverse Action process.* Status

Bugs

- **Issue**: Section 32 caluclation using wrong index value.
  Response: Obtained additional legal opinions and recoded the index value choice routing. Section 32
- **Issue**:
  Response:
- **Issue**:
  Response:
-

Enhancements

- **Issue**: Fees can be based a percentage of either Gross Loan, Base Loan, Sales Price or Appraised Value.

Response: Enabled this option in the system.

- **Issue**: Impound payment periods do not allow defining which items have been paid and what year they may be paid in.
  Response: Overhauled the definiion of defining the payment period to include year and also allow setting an impound item to have been paid (refi's). This enhances the flexibility of defining inpound items. The combo boxes for the date select are auto populated based on the Interest From date and will detect changes to the interest From date and recalculate a new date range.
- **Issue**: Sometimes system may be deployed and the code may be uncompiled.
  Response: Put is a test at logon that will send a message to the user that the system is not compiled. Also in a loan file a red "X" will appear to the left of the "Loan #" label in the upper left portion of the screen indicating that the system is not compiled. Application is uncompiled
- **Issue**: Help File.
  Response: Made the help file available to the "MtgMainMenu" by adding entry 149 in the tbl_su_openforms. The foresight.chm help file will need to be in the same directory as the MtgUser.MDB. This file should also be added to the startup data routine for deployment. Also note that this file is 9 characters long and may not be supported by an older version of novell and thus the master copy should be distributed from a file directory that supports long file names. See Help File section in 1.05 MtgUser References .
- **Issue**: Putting a loan into a final status of Cancelled/Withdrawn, Cancelled incomplete or Approved Not Accepted requires the user to make the status changes in either the UW conditions or the UW Approval sccreen.
  Response: Modified the status screen to allow these changes to be made from it and automatically add the appropriate UW conditions, update HMDA, and detemine if the Cancelled Withdrawn is being used properly. Status
- **Issue**:
- **Response**:
- **Issue**:
  Response:

2.33e 9/28/99

This release reflects a refinement of 2.33d. Additional user input was incorporated to eliminate most known bugs as well as rewrite known issues affecting performance. Stored procedures can now be used to provide data to the document print process. Most of the bugs addressed in this release are either ASC reported items or new functionality that is finally getting used by users.

SQL Database

Tables: (All of these tables are used for interface (fnma) instructions.)

- **New** tbl_int_trans_record
  New tbl_int_trans_field
- **New** tbl_int_trans

- **New** tbl_int_record_conversion
- **New** tbl_int_record
- **New** tbl_int_fill_value
- **New** tbl_int_field_processing_type
- **New** tbl_int_field_format
- **New** tbl_int_fieldtbl_su_real_estate_asset_type

Fields: (These are fields specific to fnma DU conversion data)

- **Add** tbl_ldf_pastemployment.bor_monthsonjob tinyint
  Add tbl_su_accttype.suacct_fnmacode char(2)
- **Add** tbl_su_mtgtype.fnma_mortgage_applied_for char(2)
- **Add** tbl_setup_amorts.fnma_amortization_type char(2)
- **Add** tbl_setupproperties.fnma_property_type_code char(2)
- **Add** tbl_setupoccupancy.fnma_property_will_be char(1)
- **Add** tbl_setuptransaction.fnma_loan_purpose_code char(2)
- **Add** tbl_su_improvements.fnma_purpose_of_refinance_code char(2)
- **Add** tbl_su_real_estate_asset_type.reat_id int
- **Add** tbl_su_real_estate_asset_type.fnma_real_estate_asste_type char(2)
- **Add** tbl_su_real_estate_asset_type.reat_deac varchar(35)

Stored Procedures:

- **sp_sa_seconds_233e**: NEW : Returns all the loans that are stand alone seconds
  sp_rpt_md_fu_funded: NEW : Made stored procedure from currently existing query
- **sp_qry_rpt_md_uw_approved_233e**: NEW : Need to convert qry_rpt_md_uw_approved to a stored procedure because an ODBC call failed error was occurring (BUG 951)
- **sp_qry_rpt_wrhs_info_233a**: Change <> Null to is not null in the where clause because there were no results being returned for shipped loans not sold settled
- **sp_qry_rpt_qc_funded_1**: No results were being returned due to the is not null being <> (09/29/99)

Views:

- **vw_sa_firsts**: NEW : Returns all the firsts
  vw_all_seconds: NEW : Returns all the seconds
- **vw_rpt_uw_submitdt**: NEW : Used with new stored procedure sp_qry_rpt_md_uw_approved
- **vw_rpt_uw_name**: NEW : Used with new stored procedure sp_qry_rpt_md_uw_approved

Applications
MtgUser.MDB

Update Queries to Run:

- **update_233e_addr**: NEW : Update query to update the mailing address for all non primary and non purchase loans
  update_233e_addrprim: NEW : Update query to update the mailing address for all primary and purchase loans

Bugs

- **Issue**: (bug 960) The default state schedule was not coming in under the new date

scheme.

Response: Correction to gTransferFees for new date scheme.

- **Issue**:(bug 961) Multi-select of outstanding underwriting conditions in docs stopped working (single item select only).
  Response: Correction to the class frm_MktgProg_select and support modules to enable both multi and single item select depending on where the class was called.
- **Issue**: (bug 962) Changes in loan amount would force a recalc and would reassign the mi premium to be all financed.
  Response: Correction to gUpdateMI function in fsAPR to ask the user how much to apply to amount financed whenever a recalc updates the amunt of the premium.
- **Issue**: Payment Shock in UW Worksheet.
  Response: In function mCalcPaymentShock Added a check to see if the assets value is null. If so a warning message is displayed and the division by total assets is no done.
- **Issue**: Contruction loans in New Loan Init form. The stored procedure wants a value nut form does not supply it for cinstruction loans.
  Response: In function SaveBasics added a new variable that will contain the check builder option for construction loans because the stored procedure is expecting an integer and was receiving a 'True' or 'False' value.
- **Issue**:
  Response:
- **Issue**:
  Response:

Enhancements

- **Issue**: Alaska requires Recording District and Judical district information on the note.
  Response: Added two fields on the Docs_property form for input only when state is Alaska.
- **Issue**: Users may change UW status codes and the UW options in the status screen may differ from the options in the UW_Approval form.
  Response: Changed the query so that the UW_Approval form options match what the user has identified to be in the status screen list for UW choices.
- **Issue**: Doc Inventory form users usually search by docid and that is not the default option.
  Response: Made docid the default search option.
- **Issue**: Doc Request Detail form (pop up when a subkey is assigned to a document)
  Response: defaults to joint when the document is a borrower document.
- **Issue**: Temp liability records for the 1003 item 7.
  Response: Changed from saving this record and recalcing on exit to generating the record whenever a print request is made so the temp record is not stored. This reduces the number of recalcs against the liability table.
- **Issue**: Funding Wholesale report does not accomodate MI premium for Correspondent loans.
  Response: Modified report to include the MI premium in the wire for correspondent loans
- **Issue**: Users who forget to enter interest dates will get errors messages.

Response: Added some warning messages to let the user now if the gfe screen payment dates have not been entered.

- **Issue**: form frm_property. Added a property get process for mail address.
  Response: Changed auto load of address to pick up the subject address if the propertywillbe = 1 and the purposerof loan = 1 otherwise the borrower address is used from tbl_ldf_formeraddresses.
- **Issue**: Funding form define wire/warehouse date.
  Response: Changed Wire Date to WH/Wire date, moved the field and label over and added new status bar text.
- **Issue**:
  Response:
- **Issue**:
  Response:
- **Issue**:
  Response:

Applications
MtgRpts.MDB

Bugs

- **Issue**: rpt_md_uw_approved
  Response: Changed the query behind this report from qry_rpt_md_uw_approved to a stored procedure sp_qry_rpt_md_uw_approved
- **Issue**: rpt_funding_whlsl
  Response: Changed the unbound field Date Warehoused to be bound to docs_wire_date (09/29/1999)
- **Issue**:
  Response:

2.33e 9/30/99

SQL Database

Tables: (All of these tables are used for interface (fnma) instructions.)

Applications
MtgUser.MDB

Queries:

- **qry_DocMap_BorChksRcvd:** [AmtChecks], gives a total of all the money received from the borrower toward closing.

EXHIBIT B

Doc #2

Foresight Mortgage Systems

12/15/99
- Results back from Chad

Memo

To: Gail Taylor Russell

From: Frank Tuttle

CC:

Date: 12/08/99

Re: Software Patent Search

Title: Loan Application Compliance Auditing

Business Requirement: State and Federal laws impose requirements on companies in the business of originating and closing loans secured by real estate. State laws created into law are then assigned to an agency for enforcement.

- In most states there exists licensing requirements where a company may be required to hold one or more licenses to conduct business in the state (not all states require licenses for all credit products.
- In many states licensing requirements are distinguished by type of product being offered, whether the company is a retail originator, a retail lender, a wholesale lender or a correspondent lender.
- In some states there are different rules that apply if the company has a branch office or there may exist a requirement for the company to have a branch office.
- For each license (some states have more than one agency enforcing different licenses and requirements) there are typically a set of lending requirements that the company is required to comply with. Many states impose unique restrictions not imposed by other states. Some states impose restrictions that are tighter than a Federal requirement.
- The company is subject to audits by the licensing agency of the state and faces fines, penalties and borrower reimbursements for failing to comply with the requirements (or interpretation of the requirements) in effect. Some states interpret the Federal requirements different that the Federal enforcement agency and audit the company with their interpretation of the federal requirement.

Description of Software: The software module performs an auditing function using rules stored in a database against the data in a loan file. The audits include but are not limited to:

- Provides an interface and data scheme to enter in all state licenses, exceptions and base requirements using custom rules that can be assigned to a license (as an example some licenses are for open ended loans referred to as HELOC – home equity line of credit, or closed end seconds or firsts).

- Provides an interface and data scheme to build and store a database of state specific compliance rules.
- Provides an interface and data scheme to build and store rules that can be applied to data elements in the loan file.
- Provides an interface and data scheme to assign licenses to a lender or originator in either a retail or wholesale relationship.
- Provides an interface and data scheme to assign compliance rules and specific loan rules to each state license.
- Originator license audit. That the license used to originate the loan is current and valid for the type of loan originated. There are about 6 data checks at this point.
- Lender licensing audit (if the originator is not the lender). Similar to the originator license audit but incorporates the lender license.
- Compliance audit that reads the loan data and tests the data against the compliance requirements.
- Provides an interface and data scheme to allow the configuration dictate the response to failures and how they issue the response from the compliance and license auditing functions.

Develop Patent Search results: I did not encounter any patents that perform even a remotely similar function as that outlined above. The closest I came are the following:

5361204, 5765144, 5870721, 5383113, 5611052, 5878403.

I am also not aware of any other competitor software package that has a similar function solution. U think that if there is anything out there it would reside in the legal firms that specialize in providing guidance to the industry with regards to state compliance matters. However my own inquiries with a few firms has indicated that they perform their legal work on a time basis and have not automated anything.

A Federal auditor (in the field about 15 years) has not seen anything like it.

Two state auditors have likewise not seen anything like it either.

EXHIBIT C

Taylor Russell & Russell, P.C.

Intellectual Property Attorneys and Counselors at Law

4807 Spicewood Springs Road
Building One, Suite 1200
Austin, TX 78759

Telephone: 512.338.4601
Facsimile: 512.338.4651
Internet: www.russell-law.com
gail@russell-law.com

December 16, 1999

Mr. Frank Tuttle
6 Hutton Centre Drive
Suite 275
Santa Ana, CA 92707

RE: Patent Search Results for "Loan Compliance Auditing Software and Method of Use"
Docket No. 800470

Dear Frank,

A novelty search, for a software program was performed. The software program enables companies that are in the business of originating and closing loans secured by real estate, to audit their loans' compliance with state and federal laws and regulations. Included in the search is the method of delivering the software from a web server – whether by licensing the user or having the user pay for each use.

The Current Invention

The invention provides a software program that performs an auditing function, which evaluates the compliance of real estate loans with state and federal laws and regulations. The software audits compliance with laws or regulations, which require a business to obtain certain licenses, before doing business in a jurisdiction. The software also audits compliance with rules and requirements, such as lending requirements, which are imposed upon those businesses holding each type of license. The software will also audit compliance with state-specific exceptions to federal regulations, and to the interpretation of federal regulations.

The software program utilizes an interface and data scheme, for entering and storing all types of state licenses, along with the state-specific compliance and lending rules that correspond to each type of license. Upon user request for compliance verification, the software identifies the loan type and the license(s) that is necessary for the originator or lender to extend the loan. The software identifies whether the business possesses a current and valid issue of the required license(s). The software then uses data elements in the loan file – presumably input by the user – to evaluate whether the loan complies with the state and federal requirements and exceptions, which govern the

applicable jurisdiction. The software then responds to the user, indicating the loan's compliance, or failure to comply, with the rules and requirements that govern the loan.

The invention also provides a method of delivering the loan compliance auditing software program. The method involves preparing the software program on a web server, from which users can employ the software to perform their loan compliance auditing. The software may be licensed to users for term, or on a per-use basis.

Summary of the Search Results

Based upon the search results, there are very few patents in your field of invention. As you can determine from a close review of the above enclosed patents and from what was revealed when I performed the search, there were no systems found perform an auditing function for evaluating the compliance of real estate loans with state and federal laws and regulations. A detailed explanation of the search results is discussed below.

Detailed Search Results

Patents described in sections I-III were found, by searching the patent database, for the years 1976-1999. The following patents disclosed or suggested art relevant to loan compliance auditing software, and methods of delivering such software via the web.

1) 5,765,144 System for Selecting Liability Products and Preparing Applications Therefor

The patent discloses a system for interactively interviewing and educating customers about their credit needs, and about the credit products that are available from the financial institution that is employing the system. Using information gathered from a customer, the system determines which of the available credit products would suit the customer's needs, while providing the lowest monthly payment or interest rate. The system then recommends the credit product(s) to the customer. If the user selects the recommended credit product(s), then the system generates an appropriate credit application. The system completes as much of the application as possible, before passing it on to the customer for editing. The customer may complete and edit the data on the application, via the system's graphic interface. The system then passes the completed application on to the financial institution for processing.

The patent discloses no material that is relevant to the currently invented software program. The patented system only gathers and stores data, and inserts it into a generated application. It does not manipulate the data, for the purpose of evaluating compliance with pre-set figures or logical rules.

The patent's disclosure is more relevant to the currently invented method for delivering loan-auditing software, via a web server. The patent discloses applicant access and use, of the above system, via the Internet. However, there is no "delivery" of software, to the applicant. The software that is resident on the server only queries the applicant and accepts input. The applicant cannot personally manipulate the software, or

the data that he feeds into the program, to suit his own purposes. Following from the limited use of the patented system's software, the patented method does not check whether the applicant is a term or per-use licensee of the software that is on the server.

The currently invented method allows the currently invented software program to be used and manipulated at user level, not at server level. The currently invented method is also intended to serve a potentially open-ended number of individuals – particularly in its per-use fee embodiment – via the Internet. Thus, the method necessarily involves licensing the software to users and checking each user's authority to use the software that resides on the web server. These steps distinguish the currently invented method for web-based software delivery, from the patented method.

2) 5,845,065 Network License Compliance Apparatus and Method

The patent discloses a method and apparatus for controlling devices, such as computers, in compliance with software-license restrictions. Software-licenses, which correspond to application software that is centrally located on a network server, are assigned to network users, in the order of their requests to use the applications. If no software-licenses are available, then the system may determine if software-licenses may be swapped among users to free up a software-license for the particular application requested. The network server contains a database, whose information is used to perform all analyses.

To exemplify the patented invention, suppose that MS Office is placed on the network server, in addition to individual copies of MS Word and MS Excel. Now suppose that 2 software-licenses for MS Word, 1 software-license for MS Excel, and 1 software-license for MS Office, have been granted to the network, as a whole. The server will take requests to use the three applications, from network users. The server will only allow two users to use MS Word, at a time, on a first-come first-served basis. Similarly, the server will only allow one user to use each of MS Office and MS Excel, at a time.

If User1 and User2 request the use of MS Word, they will be assigned to the network's two software-licenses for MS Word. They will then be granted use of the application. If User3 requests the use of MS Office, in order to use Excel, he will be assigned to the network's only software-license for MS Office. User3 will then be granted use of MS Office. A network user, who subsequently requests the use of Office or Word, must wait until one of the current users is finished. However, in one embodiment of the patented invention, the network may swap software-licenses. If User4, for example, requests the use of MS Word, the network server may assign the available MS Excel software-license to User3, and assign the MS Office software-license to User4. In this way, User4 can use the MS Word application (contained in MS Office), without waiting.

The patented invention may also compare network user information, against restrictions that are in the software-licenses that have been granted to the network. For example, if a software-license for MS Word states that it may only be used by network users in a certain geographical location, then the server will analyze each next user who requests MS Word. If the requesting user is not within the specified locale, then the

server will deny the user's request and perform the analysis on the next user who has requested MS Word.

The apparatus disclosed by the above patent is only superficially relevant to the currently invented loan compliance auditing software. An invention, which analyzes whether a specific instance of software usage complies with software-licenses, may be analogous to a software program, which analyzes whether a specific instance of lending complies with applicable governmental regulations. However, the analogy only holds up under such broad characterizations. Fundamentally, the patented system differs, because it automatically performs compliance verification, using a program on the network server. Whereas, in the current invention, verification is performed by users via the software that they request from the server.

Additionally, the patented system only controls the number and characteristics of a pre-set group of individuals, who simultaneously use a centrally located software application. Thus, the patented system is much simpler – almost resembling a switch. The currently invented software program, on the other hand, chooses from a broad expanse of state and federal regulations, to audit the many data elements contained in a specific loan file. Thus, the software program is necessarily apprised of sophisticated logic that allows it to choose among many regulations – which are often both quantitative and qualitative – and to translate the regulations into logical rules that can be applied to raw data.

Hence, the little that is disclosed in the patent, about *how* the patented apparatus analyzes compliance with software-licenses, is not sufficient to even suggest its use outside of a single network. So much less, does it suggest using the apparatus' analysis technique, for a sophisticated purpose such as auditing loan compliance.

The patent discloses a method that is relevant to the currently invented method of delivering a software program, via the web. The patented method involves storing software applications on a server. The method also involves maintaining software-license data on the server. The patented method involves processing users' requests for the software that resides on the server, and limiting usage of the software applications that are on the server, according to the software-license data that is also on the server. When users are granted use of a software application, they may manipulate the software at user level, not server level. The currently invented method, too, involves maintaining software on a server, and determining whether those who request use of the software may be granted usage. Also like the patented method, the currently invented method allows user manipulation of the software that they request, at user level.

Despite some similarities, though, the currently invented method differs significantly from the patented method. The patented method involves delivering software over a single network, thereby limiting the number of possible users. Likewise, the patented method need never involve examining whether users possess individual licenses to use the software that is located on the server. Conversely, the currently invented method involves maintaining software on a web server. Thus, it may serve an open universe of potential users. Consequently, the currently invented method necessarily involves checking whether each user, who makes a request to use the compliance verification software, is individually licensed to use the software program. This requires more interaction, than the patented method, with users who are requesting

software usage. Additionally, the currently invented method requires the step of analyzing whether each user possesses an individual license to use the software program.

3) 5,680,329 Fire Protection Code Compliance Verification System and Method

The patent discloses a system that is designed to help ensure compliance with fire-protection testing and maintenance standards, which are set forth by governing codes. The patented system utilizes sensors to acquire data about a building's fire protection system, each day. The system then uses the raw data to determine whether the fire protection system complies with applicable fire protection codes. Finally, the system prints a periodic compliance verification report, which can be electronically transmitted to the party that is charged with the care of the building.

The patented system uses sensors that are arranged within a fire protection system, such that the sensors measure parameters that are essential to the fire protection system's operability. One example might include arranging sensors to measure output voltage, at the terminal connections of any batteries in the system. Another example might include arranging sensors along ceiling sprinkler pipes, to measure the water pressure inside the pipes. Sensors gather data each day and feed their data to a recorder. The recorder stamps the data with day and time, and then sends the data by modem, or other connection, to an off-site computer for archiving.

After a set interval of time has elapsed – a month, for instance – the off-site computer evaluates the gathered data, in order to determine if the fire protection system meets governing codes. The patent discloses that this evaluation may proceed, according to specific software, query logic, Boolean logic, flag setting, and/or computational comparisons of data with pre-determined benchmark values. Once the computer has evaluated whether the fire protection system complies with government standards, the computer electronically forwards a report to the parties responsible for the building's fire protection. Where problems exist in the fire protection system, the computer provides for real-time notification.

The patented system is also capable of evaluating the adequacies of many different types of fire protection systems. The patent discloses that separate logic is used to evaluate compliance, in each type of fire protection system, because fire code standards vary among types of fire protection systems. Hence, the patented system applies a specific set of rules for each fire protection system that it evaluates.

The patented system performs the same type of function that is approached by the current invention. In the patented invention, a remote computer system uses a program to evaluate whether a fire protection system complies with the government regulations that govern the specific location and fire protection system. The patented system utilizes sophisticated logical rules, in order to apply government regulations to raw data elements that are gathered from the fire protection system. The patented system then reports compliance or non-compliance to the appropriate parties.

The currently invented software differs from the patented system in four ways. Two are significant, and two are not. First the purposes for which the compliance verification software is used, in each invention, is different: fire protection, versus lending. However, the difference is not likely to be significant to patentability. The

purpose for which the currently invented software has been designed is only as significant as the unique features which the purpose necessitates. These features include translating governmental regulations into logical rules that can be applied to raw data elements, and providing many sets of rules that correspond to differences in governing regulations. The patented invention provides the translation of government codes into logical rules, albeit in a different field. Additionally, the patented invention provides multiple sets of logical rules, which correspond to variations in governing fire codes. Hence, the features required by the current invention's different purpose are provided by the patented system. Consequently, that purpose is unlikely to make the currently invented software patentable, on its own.

The second difference between the inventions is that users of the currently invented software must input loan file data to the software, so that it can be evaluated for compliance with applicable regulations. The patented system automates the data input process. This difference is unlikely to cause the currently invented software to be patentable, over the patented system, because it is the patented system, which goes the extra step in data input. Thus, the mere difference of manual input would likely render the currently invented software obvious, in light of the patented system.

Similarly, users of the current invention must request use of the invented software program for each compliance verification. The patented system automates this process, also, by performing compliance verification at pre-set intervals. The user of the patented system need only, effectively, make a single request for all subsequent compliance verifications. Unlike the difference in data input, though, the difference in requesting compliance verification may be significant enough to promote patentability.

The patent does not disclose the ability of users to request verification compliance outside the pre-set time interval. There are increased advantages and system requirements that are necessarily present, when code compliance can be verified on demand. Hence, this points to one advantage of claiming the present invention as a system and method, rather than a software program and method. The current invention could then be distinguished from the patented system, by giving users the option to request unlimited instances of compliance verification.

The final difference, between the patented system and the currently invented software, is significant, but not clear. The currently invented software automatically identifies the type of loan, whose data are to be evaluated. Hence, it also identifies the set of rules that should be applied to a specific user's request for compliance verification. This is a necessary, but not so simple task, since the software is intended to potentially serve an open-ended number of users – particularly in the per-use fee embodiment – via the internet. The patented system does not appear to have this capability. Unfortunately, though, the patent is silent on the matter of how it picks the set of rules that it will apply in a given compliance verification.

The patent suggests that the rules applied to each compliance verification are pre-set. This is inferred from several elements of the patent's disclosure. First, the only method, by which the patented system *could* automatically determine what rules to apply, in its compliance verification, is to automatically identify the type of fire protection system being used. Likewise, the patented system could only automatically identify the fire protection system type, by attempting to match the data that it receives from the sensors, with a fire protection system that contains corresponding parameters. However,

the patent provides for variation in arranging sensors in a fire protection system. The user decides where and how to arrange the sensors; and, hence, what parameters to measure. Thus, the data that is gathered from separate iterations of the same fire protection system type is unlikely to be uniform. The compliance program is unlikely, then, to identify the fire protection system type, from the data gathered by the sensors. And so, the compliance verification rules that are applied to each system must be pre-set.

The foregoing point is supported, by the fact that users of the patented system are pre-set; in essence, they are signed up for the system. Only in this way, could the reports (which are generated automatically; not upon user request) find their way to the proper parties. Thus, there is no reason for the patented system to attempt to automatically identify the user's fire protection system type. Rather, it is likely entered into the system database, along with the user's building information, and the parameters that the user chooses to measure. Corresponding compliance verification rules are likely to simply be assigned to each user, if not pre-built for each one.

Hence, the currently invented software program is likely to be patentable over the patented system, if it contains, as a necessary element, the feature of automatically identifying the set of logical rules that it will apply to a user's request for compliance verification. Alternatively, if the current invention could be distinguished from the patented system, if the current invention were also claimed as a system, which necessarily included user requests for each compliance verification.

The patent discloses no material that is relevant to the currently invented method for delivering a software program via the web.

4) 5,995,947 Interactive Mortgage and Loan Information and Real-Time Trading System

The patent discloses a method and system for trading loans in real time, by producing loan applications and placing them up for bid, by potential lenders. The system employs a transaction server, which maintains a database of pending loan applications. The transaction server is connected to stations, via the Internet or other means. Each station corresponds to its user's role in a loan arrangement. For instance, there can be separate broker stations and lender stations. Each station uses a different device (interface), by which the user can send requests to the server. These requests allow a user to search and modify database contents in a way that suits the user's role in a lending transaction. Thus, users can use the system data, to individually decide what loan applications are most attractive, and to determine how they should bid on specific loan applications. The transaction server will then accept a potential lender's bid on a loan application.

The patent is not relevant to the currently invented software. The patented system is simply a data manipulation and display system. There is no mention of comparing data to standards, to verify compliance with rules, as the currently invented software program provides.

The patented method is relevant to the currently invented method of delivering a software program. The patent discloses administering the patented system, by means of a web server that maintains the database, with which all users interact. Nonetheless, the patent does not use the web server to "deliver" its database to users. Rather, the patented

system employs a database that must show the same contents to every user, in order to effect fair bidding. Hence, the patented method must (and does) only allow users to request that *the data server* manipulate the *display* of its data in a certain manner. The patented method, then, only rises to the level of providing centralized browsing of database contents. Hence, it is not surprising that the patented method does not involve determining whether users have obtained software-licenses or paid per-use fees, before allowing users to take advantage of server contents.

In contrast, the currently invented method delivers software from a central *source*, so that the software may be manipulated at user level (rather than at server level). In this way, users can evaluate loan file data, in reference to regulations that are built specifically for each of their varying jurisdictions. Following from the currently invented method's delivery of loan auditing software, via the web, the method involves checking whether each individual, who requests use of the software program, is licensed to use it or has paid a per-use fee. The currently invented method can be distinguished from the patented method, then, by allowing manipulation of server contents at user level, and by its checking license or per use fees, prior to an individual's use of server contents.

5) 5,966,699 System and Method for Conducting Loan Auction Over Computer Network

The patent discloses the same basic system and method, as the previous patent. This patented invention only adds the feature of allowing borrowers to submit loan applications electronically, to the database from which lenders choose. The previous patent limited bidding to loan applications already stored in the database. Thus, the patented invention, like the previous patented invention, is not relevant to the currently invented software. Additionally, while this patented invention uses a method that administers its loan auction system via the Internet, the patent discloses *no* ability of lenders to manipulate the data displayed by the web server.

6) 5,870,721 System and Method for Real-Time Loan Approval

This search would not, of course, be complete, without having examined the relevance of ever-more-common, real-time loan approval systems and methods. This particular patent discloses the use of a computer, to interface with a loan applicant. The computer obtains information about the applicant, from applicant input and from the system's connections to databases – such as credit reporting databases – via neural network. The system determines whether to approve the loan, based upon benchmark requirements and cutoffs that have been pre-set, for each data element that is gathered about an applicant. The system then forwards loan documentation to the applicant and transfers funds automatically to the applicant's account(s).

The currently invented software achieves its purpose, using the same basic approach as the patented system, even though its users are lenders and originators; not loan applicants. The loan auditing software interfaces with a user and obtains loan file data from the user. The software determines whether the data complies with state and federal regulations, by comparing the data gathered with the requirements that have been set for each data field. Finally, the software informs a user of the loan's compliance or non-compliance with applicable regulations.

each user's request for compliance verification. The greater variety and sophistication of the analytical logic that must be used in the currently invented software, and the ability of the software to automatically identify what rules to apply to a request, are each sufficient to distinguish the currently invented software from the patented system.

The patent discloses no material relevant to the currently invented method of delivering loan-auditing software.

8) 5,655,085 Computer System for Automated Comparing of Universal Life Insurance Policies Based on Selectable Criteria

The patent discloses a computer system that initiates, processes, prepares, stores, and transmits illustrations of universal life insurance. The computer system performs the functions, by accessing and reading text from a database. The patent makes a brief disclosure that is relevant to the currently invented software. The patent discloses the use of a formula, supposedly well known in the life insurance field, as the Specified Amount formula. The patent discloses that:

> "The Specified Amount formula uses Boolean logic designed to assure compliance with regulatory guidelines."

The patent does not disclose the nature of the regulatory guidelines that are served by the formula, presumably because of the assertion that the formula is well known in the field. Additionally, the patent does not disclose how the formula is used, to assure regulatory compliance. No other references are made to verifications of regulatory compliance, and the system's ability to verify regulatory compliance is not claimed.

Even if we were to give the above disclosure every favorable inference, it still would not rise to the level of the currently invented software. The Specified Amount formula is only that – a single mathematical formula. It is uniformly applied to every case. The currently invented software uses state and federal regulations to build logical rules that are specific to each user. Thus, the very purpose of the software demands more varied and sophisticated analytical logic, to verify compliance of data with applicable standards, than could even potentially be found in the patented system. Moreover, the currently invented software automatically identifies which rules to apply to each user's request for compliance verification. The greater variety and sophistication of the analytical logic that must be used in the currently invented software, and the ability of the software to automatically identify what logic to apply to a request, are each sufficient to distinguish the currently invented software from the patented system.

The patent discloses no material relevant to the currently invented method of delivering loan-auditing software.

Analysis of Search Results

In order to obtain a patent, it is necessary that your invention be "nonobvious." The standard for this is set forth in 35 U.S.C. § 103, which states in pertinent part:

> A patent may not be obtained... if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.

The patents uncovered during the search do not appear to be the same as your invention. In addition, even if the patents uncovered were combined, that combination does not appear to have the totality of advantages and features of your system. However, the PTO can always make an obviousness rejection by combining other patents not uncovered during the search or by citing existing, non-patented automated design systems. But given the very few patents in this field and the differences and advantages of your system, even if the PTO makes an obviousness rejection, it would most likely be able to be overcome.

Based upon the results of the search and your knowledge of the state of the art in your industry, I see no reason not to file a patent application for your software program. If you have any questions or comments concerning this novelty search, please let me know.

Best regards,

Gail M Taylor Russell

Gail M. Taylor Russell

EXHIBIT D

United States Patent
Tuttle

Loan Application Compliance Auditing

Abstract

State and Federal laws impose requirements on companies in the business of originating and closing loans. The laws are generally intended to protect the consumers from unethical business practices, unreasonable fees, illegitimate costs, undisclosed relationships and unfair repayment agreements. A method and apparatus is presented to automate the determination of State and Federal compliance through the application of rules, tests and calculations on a set of data that represents an application for credit (loan). The loan audit can be performed before a loan is closed to insure that the lender and/or originator have complied with all the State and Federal lending laws and thus not be liable for potentially significant violations resulting in fines, penalties, reimbursements and loss of licenses. The loans can also be audited after closing (including but not limited to):

1. on behalf of the borrower.
2. on behalf of an investor prior to sale or transfer.
3. on behalf of a pooling or rating agency who verify quality for securities.
4. on behalf of the State and Federal auditing agencies.
5. on behalf of the lender as a statistical quality control program.

In order to initiate an audit the loan application data must be presented to the audit engine that sequentially reads all applicable rules that must be enforced and then performs the checks and calculations against the data from the loan file. There are a number of different methods to expose the loan data to the audit engine including but not limited to:

1. Fax the appropriate data to a provided point where the loan data is manually entered into the audit system.
2. Enter the data via data capture interface screen connected through the Internet.
3. Provide the loan data in a well structured format file (flat file) and then deliver the file via the internet, email, ftp, floppy disk or some other delivery vehicle.
4. Obtain a license for the auditing system and run it on premise.

The results of the loan audit will be immediately available with textual descriptive explanations for the findings. The loan audit system includes the user interface and rules building processes

so that an unlimited number of rules and tests can be added to the library of available rules to use in the auditing. The system also captures an unlimited number of State licenses and allows for a unique set of rules to be related to each license and thus applied.

Inventors: **Tuttle, Frank D. (Irvine, CA)**

Assignee: **Foresight Mortgage Systems, LLC.** (Santa Ana, CA)

Appl. No.: **xxxxxx**

Filed: **xxxxxx**

U.S. Class:
Intern'l Class:
Field of Search:

References Cited

Foreign Patent Documents

Other References

Parent Case Text

Claims

Description

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to credit transactions regarding the lending of money. More specifically, the present invention is a method and apparatus for auditing State and Federal laws automatically as they apply to loan application data.

2. Discussion of Background

Granting credit and borrowing money are synonymous with lending and that business of lending dates back many hundreds if not thousands of years. Throughout the history of lending there have been many lessons learned that have been passed on in the way of laws that serve to protect the parties involved in credit transactions. A number of laws are in effect to protect the borrower and to establish transactional standards and requirements. The Federal government as well as individual States have enacted laws and posses governing agencies to enforce the compliance of these laws. The resulting laws provide the guidance and requirements for acting in the capacity of a lender within the various jurisdictions involved.

Inclusive in these laws are clearly defined requirements for a consistent, complete and timely disclosure of transactional details. The requirements include but are not limited to costs and fees, the parties involved and their relationships, the cost of credit using standardized calculations, limits and restrictions on repayment details, limits and restriction on penalties and many other issues. The requirements are varied and complex. So complex that the only method that has been used to verify a business's compliance to the requirements is a tedious manual operation involving a person looking through all of the data in a loan file and trying to find and identify any violations. This has been performed on a statistical sampling basis because of the amount of time it takes to audit an individual loan. Not only is this process subject to the hit or miss proposition of sampling, it is also subject to varying degrees of individual biases in interpretation of their own understanding of the requirements and their personal limitation for calculations and complex relationships as described in descriptive terms. The current process is inefficient, time consuming, subject to the individual biases of the auditor, the process is non standardized and also subject to the problems of small sampling.

SUMMARY OF THE INVENTION

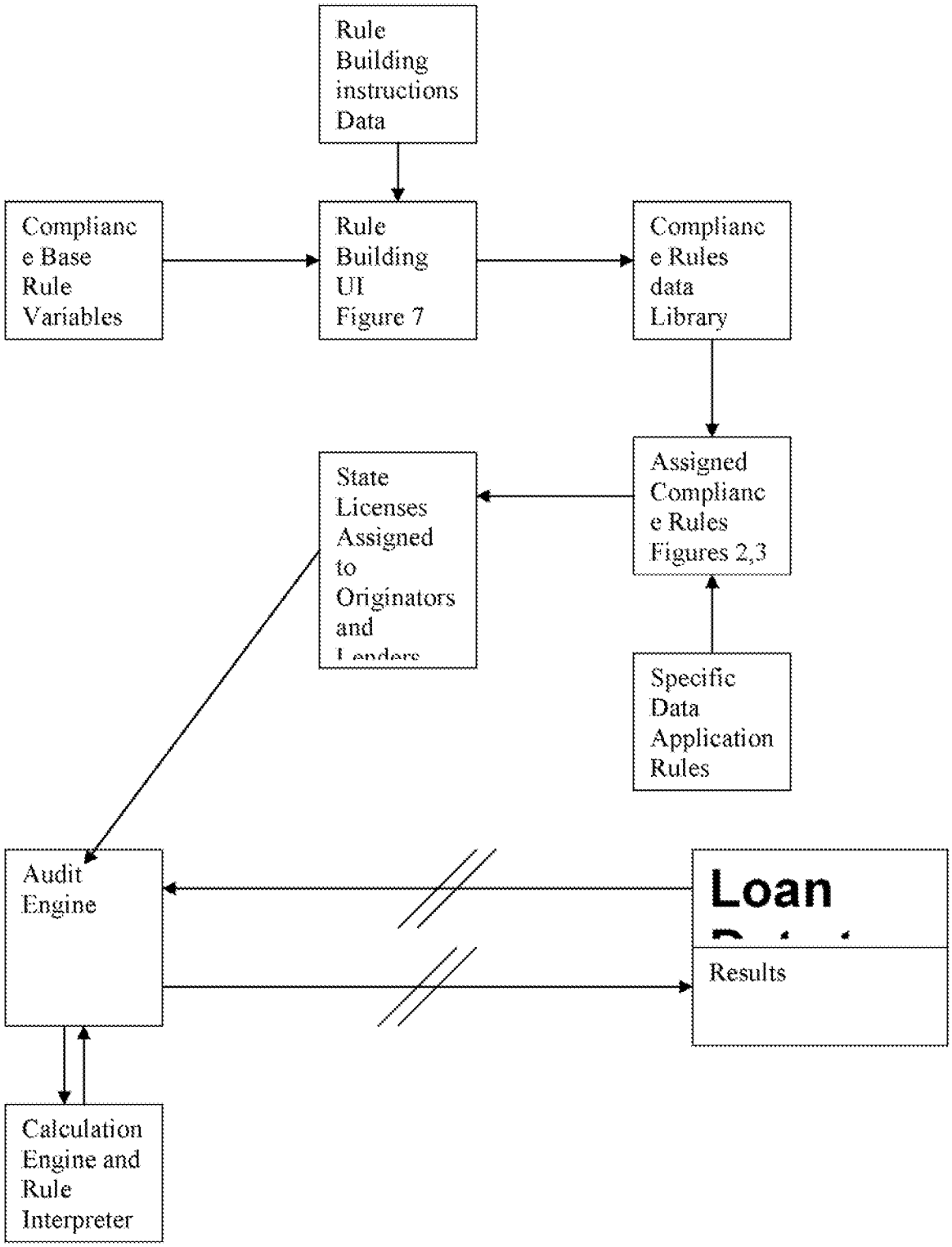
BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings,

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Loan Application Compliance Auditing

Figure 1. Componenet Relationships



Explanation of block diagram elements.

Compliance Base Rule Variables (See the "Data Field" in Figure 7): The base rule variables are the elements that the rules are constructed from. The base rule variables represent data elements in a loan file. Items such as prepaid interest or sum of broker fees can be represented by a base variable.

Rule Building and Rule Building Instructions: Base variable are assembled in math equations using the operands (+, -, /, *, >, <, <=, >=, In) to represent a State or Federal requirement or restriction. The rule building user interface as shown in figure 7 allows users to select the desired base variables and the appropriate operands to assemble the desired rule. The user interface provides the rule building instructions inherent in its design and controls the rule building process to eliminate errant rules due to math equation construction and math operator errors.

Compliance Rules Data Library: This library is the database of compliance rules that have been built. Rules need only be built one time and stored in this library and then they can be referenced to many licenses in another process step. The rules library user interface is shown in Figure 5. Clicking on a rule description in the left list box display will show the detail rule equation in the main body of the user interface. A new rule can be added by double clicking in the large text area that calls the rule builder (Figure 7). After building a rule with the rule builder the rule equation is transferred back into this large text box and can be saved with an understandable textual description.

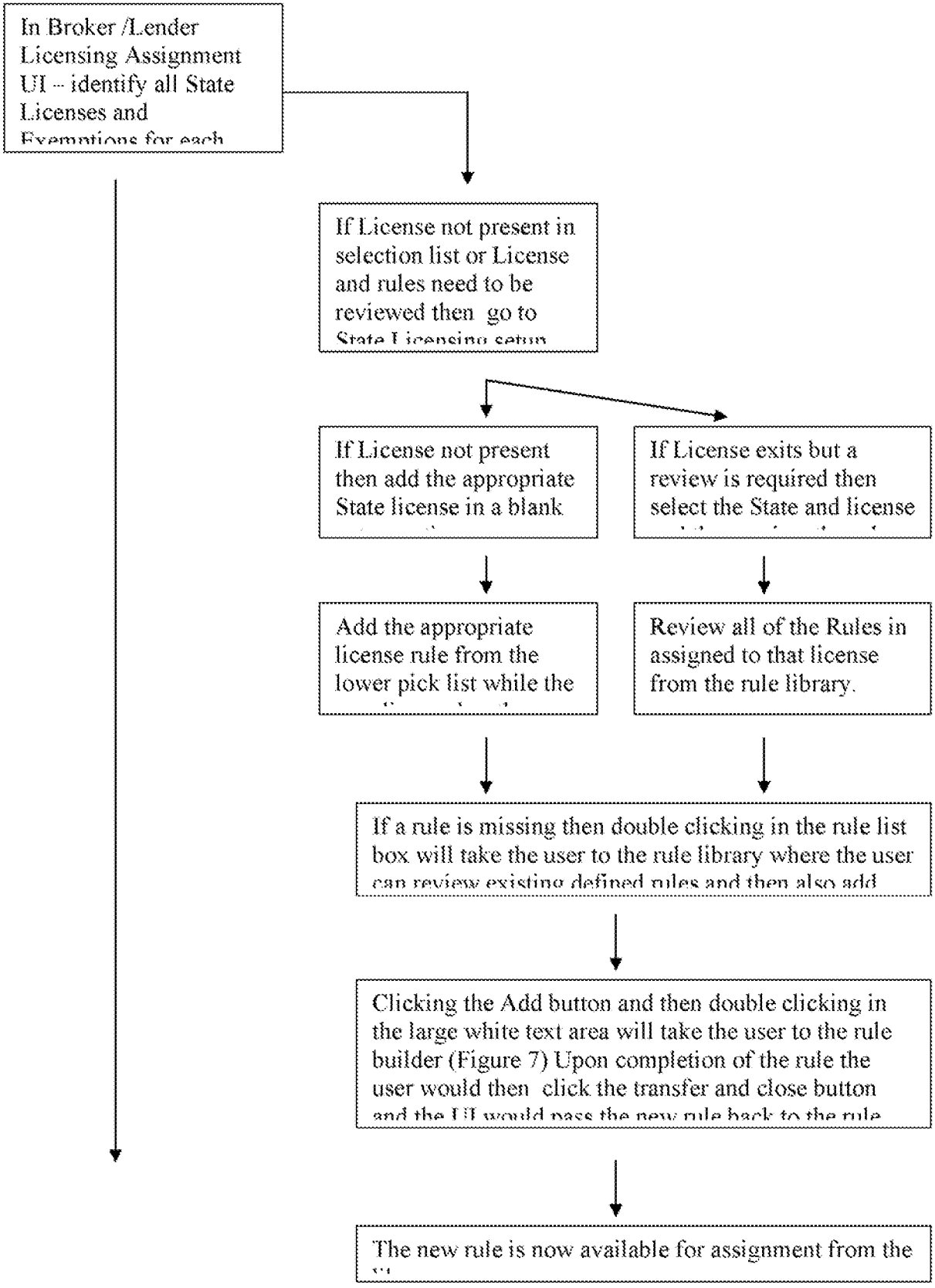
State License Building and Assigned Compliance Rules (Figures 2, 3 & 4): The "Licensing" section of Figure 3 shows where the All State Licenses are entered and maintained. Each State may allow one to many different licenses to be used to originate loans on properties within its boundaries. The Licensing section allows the entry of the specific data regarding the State authoritative agency as well and details about contact for that license. The lower section on the same tab identified as "State Rules and Limits" allows for the appropriate rules to be related to the specific license. By giving the focus on a license in the upper section, one to many rules can be associated with it in the lower section. Giving the focus to a different license will result in another unique set or combination of rules to be related to it. Figure 4 shows an expanded combo list box of the available rules from the rules library that can be selected from in order to add a rule to a specific license. The "Loan Rule" (Figure 6 is the library) field in this lower section allows for greater differentiation as to when rules may be applicable for a given license. For instance a rule may only be applicable for a first mortgage and a different rule may be in effect if the mortgage is a second. The "Loan Rule" field allows the greatest possible application control over how and when rules are applied that allows this system to adeptly manages and apply the proper rules. Just like the State Limits, the loan rules can also be assigned as needed to the State Limits from a master library of loan rules.

State Licenses Assigned to Originators and Lenders (Figure 8): Figure 8 shows the Originator and Lender setup user interface screen where there is a place to assign the license that the entity is using in each State. In the case where there is a Federal exemption and the State license is not required or is not applicable, the user would identify the proper reason for the exemption. Certain entities like Federally Chartered Credit Unions of Banks are exempt from State requirements.

These exemptions may introduce a unique but standard set of requirements into the auditing process.

At this point the setups are covered by the above explanations. When a loan file is introduced to the audit engine, the engine will read a specific set of data about the licensing requirements from the application file and get the applicable requirements from the rules that have been related to the licensee in effect If any of the rules that are applied fail then a failure response will be generated that identifies the specific reasons for the failure. The calculation engine interprets the rules that need to be applied and then performs the appropriate data math operations on the loan file data and then presents its findings for the specific rule evaluation back to the audit engine that is managing and coordinating the audit process.

Flow chart of typical events performed to assign rules.



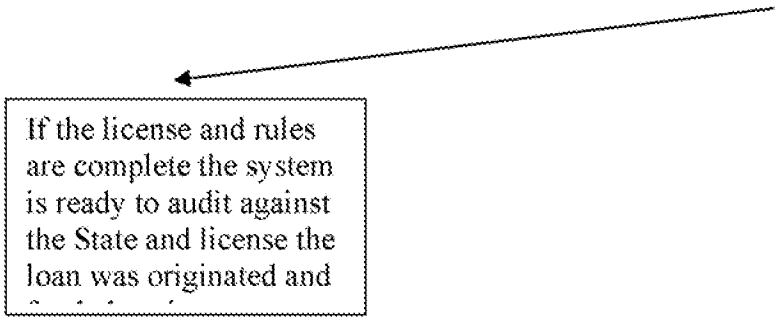

If the license and rules are complete the system is ready to audit against the State and license the loan was originated and

Figure 2. This is the main UI screen for the State specific data.

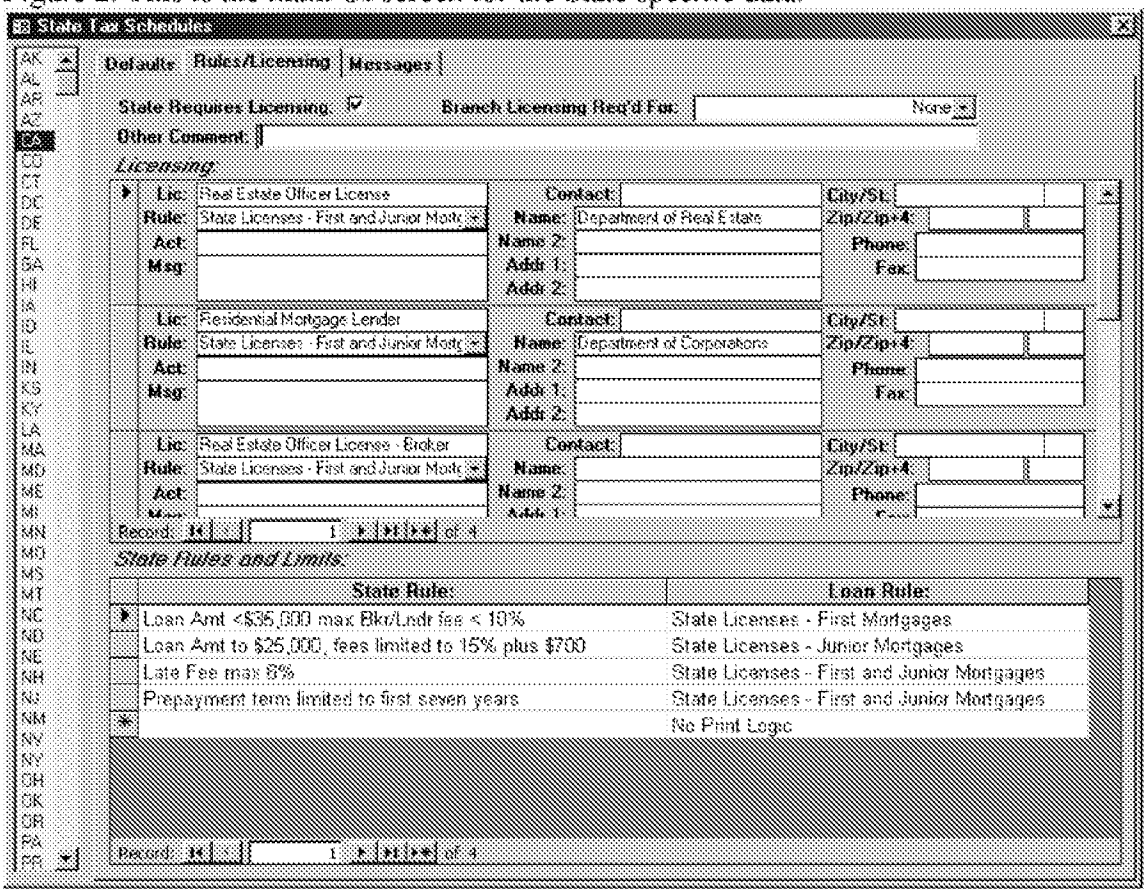


Figure 3. This is the Rules/Licensing tab UI for the State specific data setup. On this screen the available licenses for each State are defined (Licensing). For each license defined in a State specific requirements in effect are defined below. Requirements are defined as a rule using the rule building tools and then the rules are selected from a pick list and used wherever they apply. Figure 3 shows a portion of the rules list that is displayed using a combo list box control.

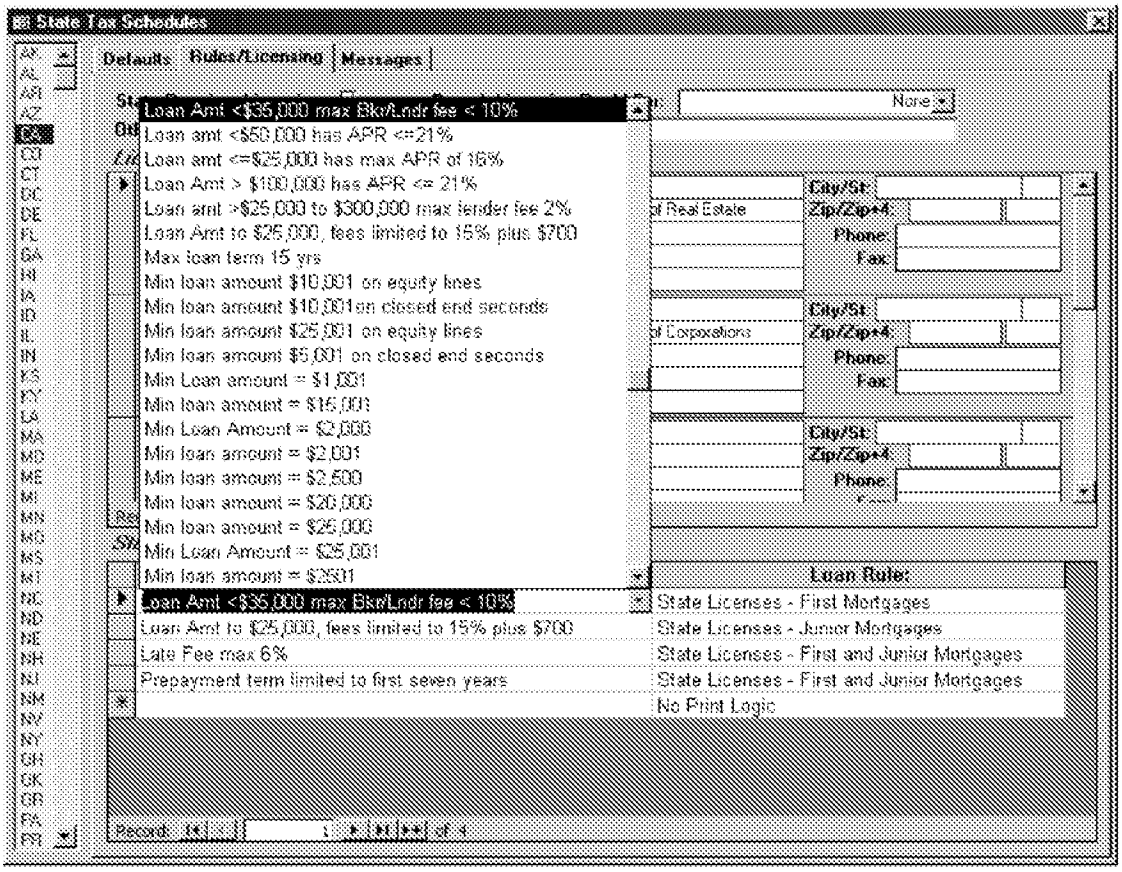


Figure 4. See above.

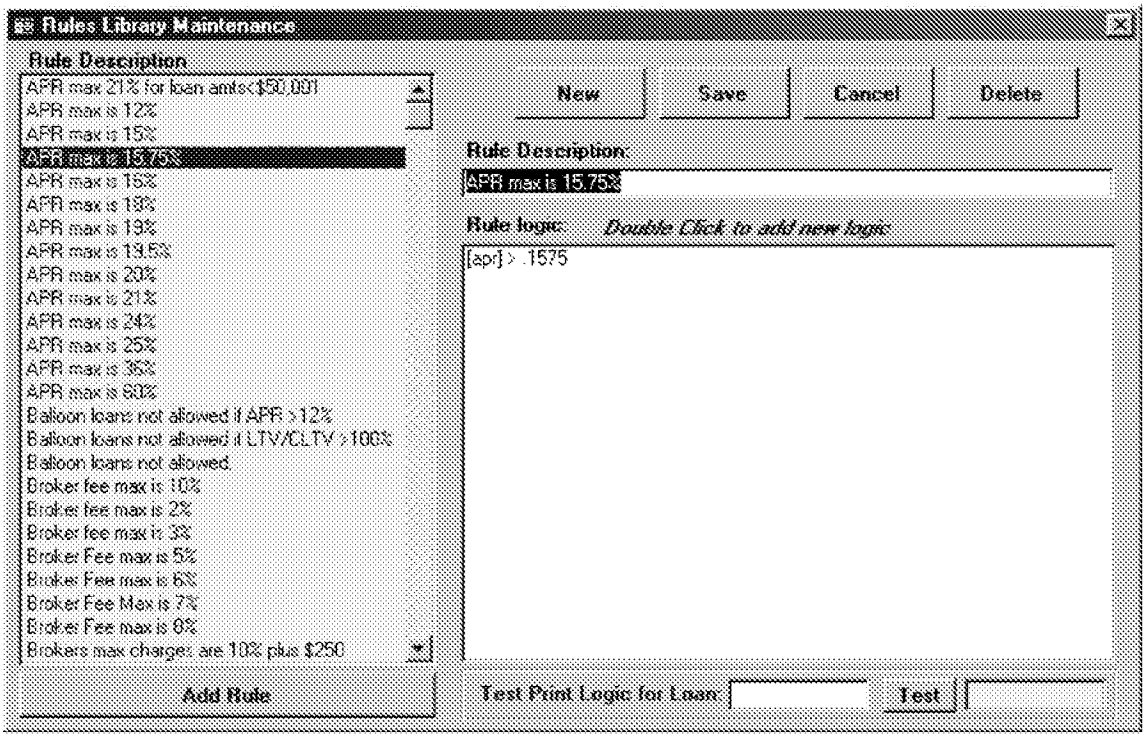


Figure 5. This is the object that displays the rules library and allows for manually modifying a rule.

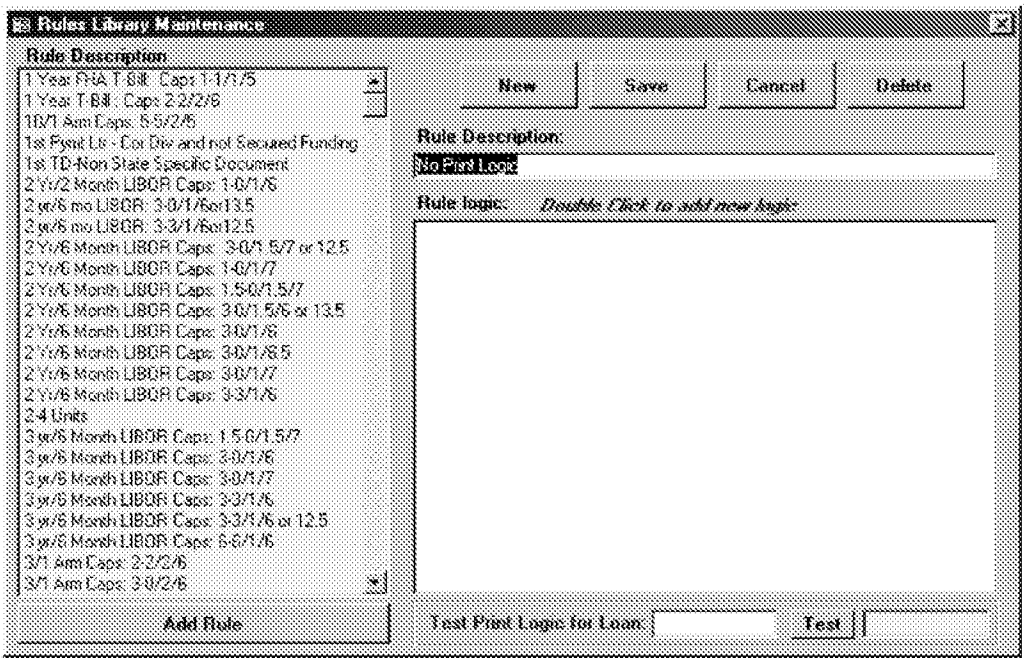


Figure 6. This is the same library object that is being used to display and manage another set of rules used in the auditing process.

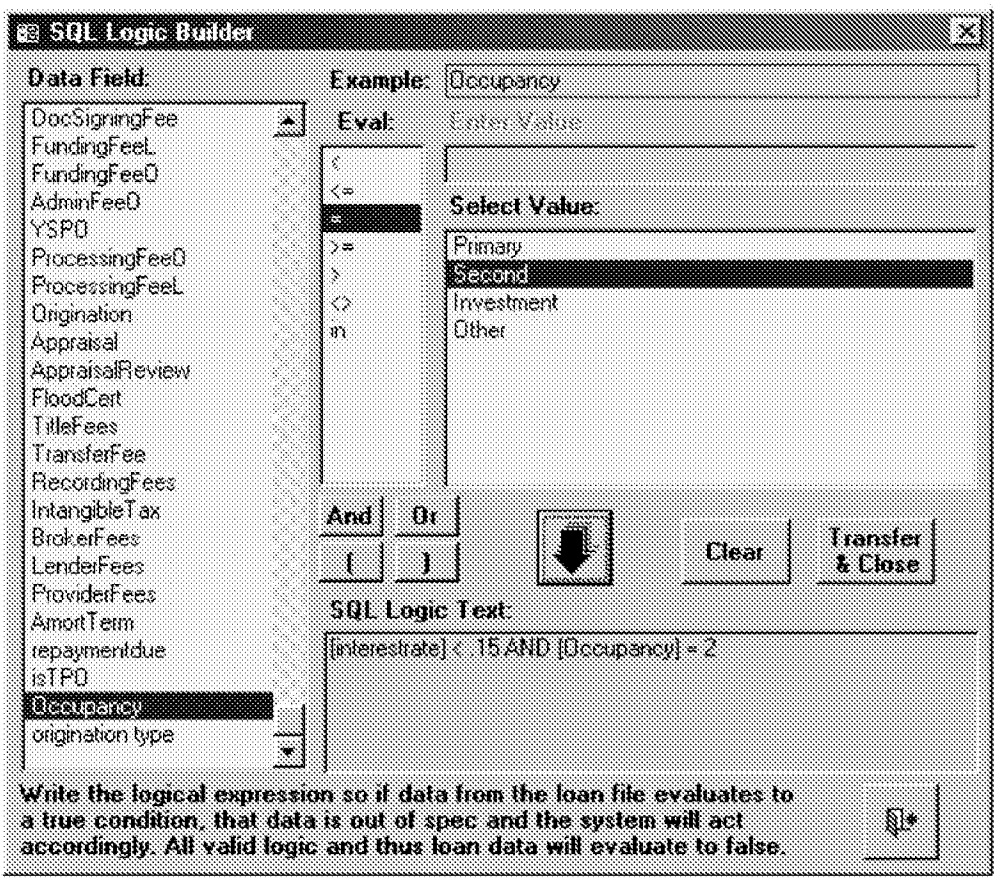


Figure 7. This is the UI used to automatically build rules for the State and Federal compliance library.

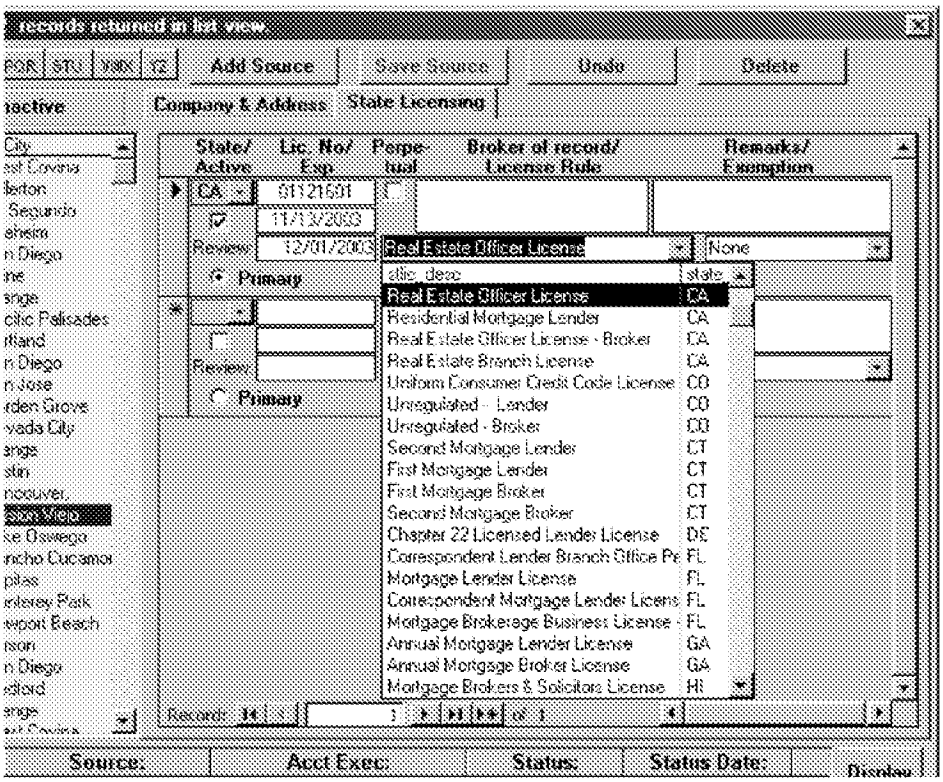


Figure 8. This is the UI screen used to assign licenses to originators and lenders. The combo list box is open to display the library of licenses that are available and can be assigned.